AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1-33. (canceled).

34. (currently amended) A method of software change modeling of networked nodes on a computer system, the method comprising the computer-implemented steps of:

~~providing a software update simulator on said computer system;~~

simulating, using a software update simulator on a computer system, processes from at least one node of the networked nodes ~~on said computer system~~;

wherein each ~~functional~~ simulated process ~~that is simulated~~ is a minimal version of a functional process that runs on said node; and

receiving a software update for said node by said software update simulator;

wherein the software update contains ~~a software package or~~ a set of one or more software packages;

wherein [[a]] each software package of the set contains at least one software module with corresponding software dependency information;

wherein said software update simulator notifies a control process for said node that a software update is being requested; and

wherein said software update simulator passes said control process identities of the set of one or more software ~~package(s)~~ packages to be updated and software dependency information.

35. (original) A method as recited in Claim 34, wherein said control process determines running functional node processes that will be affected by the software update using the software dependency information.

36. (original) A method as recited in Claim 35, wherein said control process notifies processes that have indicated interest in software updates that the software update is being requested; wherein each notified process evaluates the effect that the software update will have on its operation; wherein if any of the processes determine that the software update will degrade or have a negative impact on said node's normal operation, the process returns a veto to said control process along with reasons why; and wherein if a process finds that the software update will have no negative effects, the process returns an acceptance of the software update to said control process.

37. (original) A method as recited in Claim 36, wherein said control process waits for all of the notified processes to return the results of their evaluations and once all of the processes have reported to said control process, said control process notifies said software update simulator if any of the processes have vetoed the software update along with their reasons.

38. (currently amended) A method as recited in Claim 37, wherein said software update simulator displays node identifiers and the processes that have vetoed the software update along with their reasons to [[the]] a user.

39. (original) A method as recited in Claim 34, wherein a user initiates a software update by loading an image containing the software update into said software update simulator.

40. (currently amended) A method as recited in Claim 39, wherein the user indicates what nodes and which of the set of one or more software ~~package(s)~~ packages are to be updated.

41. (original) A method as recited in Claim 34, wherein a software package contains version information, dependency information, and other metadata information pertaining to software in the package.

42. (original) A method as recited in Claim 41, wherein the metadata includes a list of application program interface (API) providers and consumers.

43. (currently amended) A method of software change modeling of nodes in a network of nodes on a computer system, the method comprising the computer-implemented steps of:

~~providing~~ executing a software update simulator on said computer system;

wherein said software update simulator runs software components normally run on a master node in the network of nodes;

~~wherein~~ receiving a ~~user loads a node's~~ current software configuration of a node into said software update simulator by ~~loading~~ receiving current software modules installed on [[a]] said node;

~~wherein the user requests~~ receiving a request for a simulation of a software update by ~~loading~~ receiving an updated software image into said simulator;

wherein the software image contains ~~a software package or~~ a set of one or more software packages;

wherein [[a]] each software package of the set contains at least one software module with corresponding software dependency information;

wherein said software update simulator calculates the software update's impact on said node using [[the]] a current software configuration of said node; and

displaying the calculation's results to [[the]] a user.

44. (currently amended) A method as recited in Claim 43, ~~wherein the user also indicates to~~ further comprising receiving at said software update simulator the type of node being analyzed.

45. (original) A method as recited in Claim 43, wherein said software update is a software downgrade where modules are being removed.

46. (currently amended) An apparatus of software change modeling of nodes in a network of nodes on a computer system, comprising:

a software update simulator on said computer system;

wherein said software update simulator runs software components normally run on a master node in the network of nodes;

~~wherein a user loads a node's~~ means for receiving a current software configuration of a node, in the network of nodes, into said software update simulator by ~~loading~~ receiving current software modules installed on [[a]] said node;

~~wherein the user requests~~ means for receiving a request for a simulation of a software update by ~~loading~~ receiving an updated software image into said simulator; and

wherein the software image contains ~~a software package or~~ a set of one or more software packages;

wherein [[a]] each software package of the set contains at least one software module with corresponding software dependency information;

wherein said software update simulator calculates the software update's impact on said node using the current software configuration of said node; and

means for displaying the calculation's results to [[the]] a user.

47. (currently amended) An apparatus as recited in Claim 46, ~~wherein the user also indicates to~~ further comprising means for receiving at said software update simulator the type of node being analyzed.

48. (original) An apparatus as recited in Claim 46, wherein said software update is a software downgrade where modules are being removed.

49. (currently amended) A computer-readable storage medium carrying one or more sequences of instructions for software change modeling of nodes in a network of

nodes on a computer system, which instructions, when executed by one or more processors, cause the one or more processors to carry out the steps of:

~~providing~~ executing a software update simulator on said computer system;

wherein said software update simulator runs software components normally run on a master node in the network of nodes;

~~wherein~~ receiving a ~~user loads a node's~~ current software configuration of a node into said software update simulator by ~~loading~~ receiving current software modules installed on [[a]] said node;

~~wherein the user requests~~ receiving a request for a simulation of a software update by ~~loading~~ receiving an updated software image into said simulator;

wherein the software image contains ~~a software package or~~ a set of one or more software packages;

wherein [[a]] each software package of the set contains at least one software module with corresponding software dependency information;

wherein said software update simulator calculates the software update's impact on said node using [[the]] a current software configuration of said node; and

displaying the calculation's results to [[the]] a user.

50. (currently amended) A computer-readable storage medium as recited in Claim 49, wherein ~~the user also indicates to~~ the one or more sequences of instructions include instructions which, when executed by the one or more processors, further cause the one or more processors to carry out the steps of receiving at said software update simulator the type of node being analyzed.

51. (previously presented) A computer-readable storage medium as recited in Claim 49, wherein said software update is a software downgrade where modules are being removed.

52. (currently amended) An apparatus for software change modeling of networked nodes on a computer system, the apparatus comprising:

~~a software update simulator on said computer system;~~

means for simulating, using a software update simulator on a computer system, processes from at least one node of the networked nodes ~~on said computer system~~;

wherein each functional process that is simulated is a minimal version of a functional process that runs on said node; and

means for receiving a software update for said node by said software update simulator;

wherein the software update contains ~~a software package or~~ a set of one or more software packages;

wherein [[a]] each software package of the set contains at least one software module with corresponding software dependency information;

wherein said software update simulator notifies a control process for said node that a software update is being requested; and

wherein said software update simulator passes said control process identities of the set of one or more software ~~package(s)~~ packages to be updated and software dependency information.

53. (currently amended) A computer-readable storage medium carrying one or more sequences of instructions for software change modeling of networked nodes on a computer system, which instructions, when executed by one or more processors, cause the one or more processors to perform:

~~providing a software update simulator on said computer system;~~

simulating, using a software update simulator on a computer system, processes from at least one node of the networked nodes ~~on said computer system~~;

wherein each functional process that is simulated is a minimal version of a functional process that runs on said node; and

receiving a software update for said node by said software update simulator;

wherein the software update contains ~~a software package or~~ a set of one or more software packages;

wherein [[a]] each software package of the set contains at least one software module with corresponding software dependency information;

wherein said software update simulator notifies a control process for said node that a software update is being requested; and

wherein said software update simulator passes said control process identities of the set of one or more software ~~package(s)~~ packages to be updated and software dependency information.

54-57. (canceled).

58. (currently amended) An apparatus comprising:

a software update simulator on a computer system;

one or more processors;

one or more sequences of instructions which, when executed by the one or more processors, cause the one or more processors to perform:

simulating processes from at least one node on said computer system, wherein each functional process that is simulated is a minimal version of a functional process that runs on said node; and

receiving a software update for said node by said software update simulator;

wherein the software update ~~a software package or~~ a set of one or more software packages;

wherein [[a]] each software package of the set contains at least one software module with corresponding software dependency information;

wherein said software update simulator notifies a control process for said node that a software update is being requested; and

wherein said software update simulator passes said control process identities of the set of one or more software ~~package(s)~~ packages to be updated and software dependency information.

59. (previously presented) An apparatus as recited in Claim 58, wherein said control process determines running functional node processes that will be affected by the software update using the software dependency information.

60. (previously presented) An apparatus as recited in Claim 59, wherein said control process notifies processes that have indicated interest in software updates that the software update is being requested; wherein each notified process evaluates the effect

that the software update will have on its operation; wherein if any of the processes determine that the software update will degrade or have a negative impact on said node's normal operation, the process returns a veto to said control process along with reasons why; and wherein if a process finds that the software update will have no negative effects, the process returns an acceptance of the software update to said control process.

61. (previously presented) An apparatus as recited in Claim 60, wherein said control process waits for all of the notified processes to return the results of their evaluations and once all of the processes have reported to said control process, said control process notifies said software update simulator if any of the processes have vetoed the software update along with their reasons.

62. (currently amended) An apparatus as recited in Claim 61, wherein said software update simulator displays node identifiers and the processes that have vetoed the software update along with their reasons to [[the]] a user.

63. (currently amended) An apparatus comprising:

a software update simulator on a computer system;

wherein said software update simulator runs software components normally run on a master node in the network of nodes;

~~wherein~~ receiving a ~~user loads a node's~~ current software configuration of a node into said software update simulator by ~~loading~~ receiving current software modules installed on [[a]] said node;

~~wherein the user requests~~ receiving a request for a simulation of a software update by ~~loading~~ receiving an updated software image into said simulator;

wherein the software image contains ~~a software package or~~ a set of one or more software packages;

wherein [[a]] each software package of the set contains at least one software module with corresponding software dependency information;

wherein said software update simulator calculates the software update's impact on said node using [[the]] a current software configuration of said node;

one or more processors; and

one or more sequences of instructions which, when executed by the one or more processors, cause the one or more processors to perform displaying the calculation's results to [[the]] a user.

64. (previously presented) An apparatus as recited in Claim 63, wherein said software update is a software downgrade where modules are being removed.